Parabolic inverse convection-diffusion-reaction problem solved using an adaptive parametrization

G. Deolmi[a], F. Marcuzzi[a]

[a]*Dipartimento di Matematica Pura ed Applicata, Università degli Studi di Padova, Via Trieste 63, 35131 Padova, Italy*
gdeolmi@math.unipd.it, marcuzzi@math.unipd.it

Abstract

This paper investigates the solution of a parabolic inverse problem based upon the convection-diffusion-reaction equation, which can be used to estimate both water and air pollution. We will consider both known and unknown source location: while in the first case the problem is solved using a projected damped Gauss-Newton, in the second one it is ill-posed and an adaptive parametrization with time localization will be adopted to regularize it. To solve the optimization loop a model reduction technique (Proper Orthogonal Decomposition) is used.

Keywords: Inverse problem, regularization, adaptive parametrization, time localization, Finite Element method Proper Orthogonal Decomposition

---

1. Introduction

Inverse heat or mass convection problems, classically deal with the estimation of wall heat flux densities or intensities of source terms [25, 27, 12, 24]. As mentioned in [11], inverse problems are usually mathematically ill-posed and regularization methods have been developed to ensure stable solutions. For an overview, see [33, 15, 17] for example. Classical methods are penalization such as Tikhonov's regularization [32], or Bayesian methods using prior information [3], iterative regularization [2] and regularization using singular value decomposition followed by truncation of the singular values spectrum [30].

In this paper we are interested in solving an inverse convection problem, whose direct model coincides with a parabolic convection diffusion reaction equation on a fixed domain. To deal with its ill-posedness we adopt a regularization algorithm based upon Truncated Singular Value Decomposition (TSVD) and diagonal scaling [26]; moreover an adaptive parametrization with time localization is formulated, to reduce the computational cost of the Gauss Newton algorithm and to obtain a better conditioned sensitivity matrix (cfr. e.g. figure 14).

Convection-diffusion-reaction equation can be used to model a variety of physical problems. For example in [9], this equation is used to predict water quality in rivers, by measuring the quantity of organic matter contained. The importance of these pollution is estimated by the measures of the so-called BOD (Biologic Oxygen Demand) and COD (Chemical Oxygen Demand). In [9] the problem of identifying the location and the magnitude (intensity) of pollution point sources from the measurements of BOD on a part of the river is considered: the problem of source term identifycation is solved using an algorithm based on the minimization of a cost function of Kohn and Vogelius type. Also in [29] water pollution is considered: knowing the *origin* of the source of contamination is probably the most important aspect when attempting to understand, and therefore to control, the pollution transport process. Thus, a challenging issue in environmental problems is the *identification of sources* of pollution in waters. [29] deals with source identification problem, using Boundary Elemet Method (BEM). In [11], the same problem of source estimation is considered to estimate the time-varying emission rates of pollutant sources in a ventilated enclosure, assuming that the velocity field is stationary: in fact in the frame of occupational risk prevention, the knowledge of both space and time distributions of contaminant concentration is a crucial issue to evaluate the workers exposure. Althought air pollution is considered, instead of water, the underlying model is still a convection-diffusion-reaction equation, with a different convective velocity field. In [11] source's location is supposed to be known. Possible applications of this study are concerned with cartography of pollutants in buildings, estimation of contaminant emission rates inside manufactures, leak detection, environment and process control through 'intelligent sensors' (controlled ventilation with closed-loop function of pollution threshold). A similar problem is considered in [5]. Finally in [7], a convection inverse problem is solved to determine an estimate of the source term as a funcition of the altitude and the temporal of iodine-131, caesium-134 and caesium-137 in the Chernobyl disaster.

In general, in inverse convection problems, either distributed control [11, 29], or boundary control [34] or both [20] are considered. In the present paper we are interested in estimated location and intensity of pollution, and we assume to deal with boundary control, i.e. we suppose that the sources are located along domain's boundary. Thus, as in [34], we deal with an inverse problem in which one is looking for the unknown conditions in part of the boundary, while overspecified boundary conditions are supplied in another part of the boundary (here the outflow region). As mentioned above, this type of problem can model both water and air pollution.

As mentioned e.g. in [13], in *inverse problems* or *optimal control* or *optimization settings*, one is faced with the need to do multiple state solves during an iterative process that determines the optimal solution. If one approximates the state in the reduced, k-dimensional space and if k is small, then the cost of each iteration of the optimizer would be very small relative to that using full, high-fidelity state approximations. Thus *Proper Orthogonal Decomposition (POD)* will be adopted in this paper as model reduction technique, to bring our study closer to a real time problem.

In section 2 the direct problem is described.In section 3, the inverse problem is formulated. Section 5 deals with the problem of known source location, while in section 6 also source position is estimated.

2. Description of the direct problem

Let $[0, t_f) \subset \mathbb{R}$ and Ω be an open, limited and Lipschitz continuous boundary subset $\Omega \subset \mathbb{R}^2$, sufficiently regular. We denote with $\partial\Omega$ the boundary of Ω. Let $C: \; [0, t_f) \times \Omega \to \mathbb{R}$, $C = C(t, \mathbf{x})$ be the solution of the following (direct) parabolic convection-diffusion-reaction equation:

$$\begin{cases} \frac{\partial C}{\partial t} - \mu \Delta C + \nabla \cdot (\mathbf{u} C) + \sigma C = 0, & in \quad (0, t_f) \times \Omega \\ C = C_0, & on \quad \{0\} \times \Omega \\ C = C_{in}, & on \quad (0, t_f) \times \Gamma_{in} \\ C = C_{up}, & on \quad (0, t_f) \times \Gamma_{up} \\ \mu \frac{\partial C}{\partial n} = 0, & on \quad (0, t_f) \times \Gamma_{down} \\ C = 0, & on \quad (0, t_f) \times \Gamma_r \end{cases} \tag{1}$$

where Γ_{in}, Γ_{up}, Γ_{down} and Γ_r are given disjoint sets such that $\partial\Omega = \Gamma_{in} \cup \Gamma_{up} \cup \Gamma_{down} \cup \Gamma_r$.

Suppose that $C_{in} \in H^{\frac{1}{2}}(\Gamma_{in})$, $C_{up} \in H^{\frac{1}{2}}(\Gamma_{up})$, the initial condition $C_0 \in L^2(\Omega)$ and the coefficients are independent on time, moreover $\mu \in L^\infty(\Omega)$, $\mu(\mathbf{x}) \geq \mu_0 > 0$ for all $\mathbf{x} \in \Omega$, $\sigma \in L^\infty(\Omega)$, $\sigma(x) \geq 0$ a.e. in Ω, $\mathbf{u} \in [L^\infty(\Omega)]^2$, $div(\mathbf{u}) \in L^2(\Omega)$ are known. The *direct problem* consists in finding the concentration C over Ω at time t_f.

As in [11], we assume that the physical properties of the fluid are constant and that the transported contaminant is considered as a passive scalar, which means that it does not affect the velocity field. Thus we suppose to know $\mathbf{u}$.

An example of the 2D domain Ω is illustrated in figure 1.

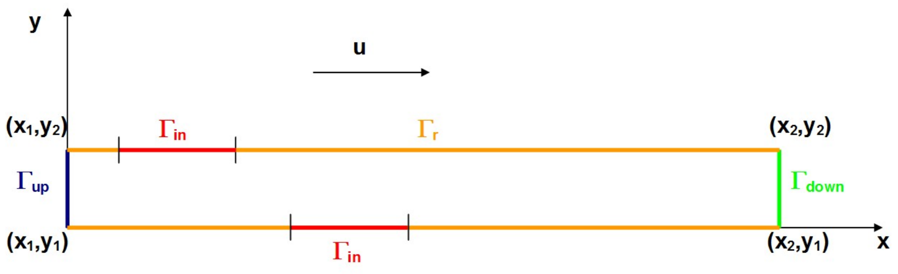

Figure 1: *Example of problem's domain* Ω*.*

2.1. Wellposedness of the direct problem and finite element discretization

Let $H^1_{\Gamma_r \cup \Gamma_{up} \cup \Gamma_{in}}(\Omega)$ be the set of $v \in H^1(\Omega)$ s.t. $v_{|_{\Gamma_r \cup \Gamma_{up} \cup \Gamma_{in}}} = 0$. Given $V \subset H^1_{\Gamma_r \cup \Gamma_{up} \cup \Gamma_{in}}(\Omega)$, the weak formulation of (1) consists in finding $C \in$

$L^2(0,t_f;H^1(\Omega)) \cap \mathcal{C}^0([0,t_f);L^2(\Omega))$ s.t.

$$\begin{array}{rcll} \frac{d}{dt}(C(t),v) + a(\mathbf{u}(t);C(t),v) & = & 0, & \forall v \in V, \\ C(0) & = & C_0, & in\ \Omega, \end{array} \tag{2}$$

where $a(\mathbf{u};\cdot,\cdot)$ is a bilinear form defined as

$$a(\mathbf{u};w,v) := \int_\Omega k\nabla w \nabla v d\omega + \int_\Omega \mathbf{u}\cdot\nabla w v d\omega + \int_\Omega \sigma w v d\omega.$$

The wellposedness of the variational formulation is studied e.g. in [28].

Consider now two families of subspaces $\{W_h,\ h>0\}$ and $\{V_h,\ h>0\}$ of $H^1(\Omega)$ and V respectively, and let $C_{0,h} \in W_h$ be a suitable approximation of C_0. Then the Finite Element (FE) discretization of (3) consists in finding $C_h \in W_h$ s.t.

$$\begin{array}{rcll} \frac{d}{dt}(C_h(t),v_h) + a(\mathbf{u}(t);C_h(t),v_h) & = & 0, & \forall v_h \in V_h, \\ C_h(0) & = & C_{0,h}, & in\ \Omega. \end{array} \tag{3}$$

Given a basis of W_h, $\{\phi_i\}$, $i=1,\dots,N_h$, where N_h denotes the number of nodes in Ω, it is well known that the FE discretization is equivalent to the solution of the following system of ODE's:

$$\begin{array}{rcl} M\dot{\mathbf{C}}(t) + A(\mathbf{u}(t))\mathbf{C}(t) & = & \mathbf{F}(C_{in}), \\ \mathbf{C}(0) & = & \mathbf{C}_0. \end{array} \tag{4}$$

where $M_{ij} = (\phi_i,\phi_j)$, $A(\mathbf{u})_{ij} = a(\mathbf{u};\phi_i,\phi_j)$ and $\mathbf{F}(C_{in})$ involves boundary conditions, in particular C_{in}.

Given a time step Δt, consider a uniform subdivision of $[0,t_f)$ s.t. $(N-1)\Delta t = t_f$. Discretizing (4) in time, using e.g. the implicit euler method, we obtain

$$\begin{array}{rcl} \left(M + \Delta t A(\mathbf{u}(k+1))\right)\mathbf{C}(k+1) & = & M\mathbf{C}(k) + \Delta t\mathbf{F}(C_{in}), \\ \mathbf{C}(0) & = & \mathbf{C}_0. \end{array} \tag{5}$$

2.2. Proper Orthogonal Decomposition (POD) reduction

To obtain a faster solution algorithm, we adopt a reduction technique. A complete overview of all classical methods can be found e.g. in [1, 31]. Since the right hand side in (4) depends on boundary conditions, it varies at each iteration of the optimization loop used to solve the inverse problem. As a consequence techniques largely used for linear constant matrices problems, like e.g. Balanced Truncation (BT), becomes too costly to be used. Thus we choose to adopt the Proper Orthogonal Decomposition (POD) method: althought its basis is stricly related to local dynamics, it is less costly to compute. In this paper we are focusing on the inverse problem solution strategy, thus we will not enter in details in the description of POD, we only summarize the main aspects: the interested reader can found a complete overview for example in [18, 14].

Given a time step $\Delta\tau > 0$ (which could be different from Δt), consider $t_m \in (0, t_f)$ and $\bar{N}$ s.t. $\bar{N}\Delta\tau = t_m$: first the unreduced model (4) is solved in $[0, t_m]$, collecting the matrix of snapshots $\mathcal{X} = (\mathbf{C}_j)$, where $\mathbf{C}_j \in \mathbb{R}^{N_h}$ is the nodal vector of the FE discretization at time $t_j = j\Delta\tau$, $j = 0, \ldots, \bar{N}$. After computing the Singular Value Decomposition (SVD) of $\mathcal{X}$, $\mathcal{X} = USV^t$, a suitable threshold k is chosen. A largely used strategy is to choose k s.t.

$$\frac{\sum_{i=1}^{k} S(i,i)^2}{\sum_{i=1}^{min(N_h,\bar{N})} S(i,i)^2}$$

is greater than a fixed tollerance. Another possibility is to impose that the first k singular values are greater than a fixed tollerance $\tau_\sigma > 0$.

It can be proved [14] that the *k-th POD basis* $\{u_i\}$, $i = 1, \ldots, k$, $u_i := U(:,i) \in \mathbb{R}^{N_h}$ is the solution of the following minimization problem

$$\min_{\boldsymbol{\xi}_1 \ldots, \boldsymbol{\xi}_k \in \mathbb{R}^{N_h}} \sum_{j=1}^{\bar{N}} \left| \mathbf{C}(t_j) - \sum_{i=1}^{k} (\mathbf{C}(t_j) \cdot \boldsymbol{\xi}_j) \boldsymbol{\xi}_i \right|^2, \qquad s.t. \qquad \boldsymbol{\xi}_i \cdot \boldsymbol{\xi}_j = \delta_{ij}, \qquad 1 \leq i, j \leq k, \tag{6}$$

i.e. for every fixed k the mean square error between the elements $\mathbf{C}(t_j)$ and the corresponding $k-th$ partial sum of $\sum_{i=1}^{k} (\mathbf{C}(t_j) \cdot \boldsymbol{\xi}_j) \boldsymbol{\xi}_i$ is minimized on average.

Finally (4) is projected on the space generated by the first k POD basis vectors, i.e. we solve the reduced system

$$\begin{aligned} U_k^t M U_k \dot{\mathbf{a}}(t) + U_k^t A(\mathbf{u}) U_k \mathbf{a}(t) &= U_k^t \mathbf{F}(C_{in}), \\ \mathbf{a}(0) &= U_k^t \mathbf{C}_0. \end{aligned} \tag{7}$$

in (t_m, t_f), where $U_k := U(:, 1:k)$, i.e. the system is projected on the subspace generated by the first k columns of U. We denote with $\tilde{\boldsymbol{C}}(t) := U_k \mathbf{a}(t)$ the estimate of $\mathbf{C}(t)$, $t \in (t_m, t_f)$ computed using POD.

3. Continuous inverse problem formulation

We are interested in solving the following inverse problem: given the additional *a priori* information

$$C = C_s, \qquad on \quad [0, t_f) \times \Gamma_{down}, \tag{8}$$

where $C_s \in \mathcal{C}^0((0, t_f), L^2(\Gamma_{down}))$ is a known scalar function, determine $C_{in}^* \in H^{\frac{1}{2}}(\Gamma_{in})$ such that

$$C_{in}^* = \arg \min_{C_{in} \in H^{\frac{1}{2}}(\Gamma_{in})} \mathcal{F}_d(C_{in}), \tag{9}$$

where the *cost function* is

$$\mathcal{F}_d(C_{in}) := \|C(C_{in}; t, \mathbf{x}) - C_s(t, \mathbf{x})\|^2_{L^2([0,t_f] \times \Gamma_{down})} = \int_0^{t_f} \int_{\Gamma_{down}} (C(C_{in}; t, \mathbf{x}) - C_s(t, \mathbf{x}))^2 d\gamma dt$$

and we have explicited the dependence of C, solution of (1), on C_{in}: $C(C_{in}; t, \mathbf{x}) := C(t, \mathbf{x})$ s.t. $C(t, \mathbf{x}) = C_{in}(\mathbf{x})$, when $\mathbf{x} \in \Gamma_{in}$.

As mentioned in [34], one may consider C_s to be a desired one. In that case, the present inverse problem is a *design problem* where the boundary flux C_{in} is controlled such that a desired concentration is achieved on the boundary Γ_{down}. C_s can also be considered to represent a continuous approximation of a set of discrete experimental *temperature measurements* obtained at finite number of locations in the boundary Γ_{down} and at discrete time instances within the interval $[0, t_f)$. In this paper we mainly refer to the second case. Observe that this class of inverse problems are of significant experimental interest for situations where the direct measurement of the heat flux C_{in} is not possible.

As indicated in [34], the main difficulty with the minimization problem (9) is the calculation of the gradient of $\mathcal{F}$. Mainly two different approaches could be used: the *first discretize than optimize* or vice versa the *first optimize than discretize*. In this paper we focus on the first strategy: in particular we adopt a discrete approximation of $\mathcal{F}^{'}(C_{in})$, combined with a Gauss-Newton approach, as explained starting from section 5.

4. Formulation of the discrete inverse problem

In the *first discretize than optimize* context, $C_s(t, \mathbf{x})$ is known only in the n_y nodes of Γ_{down}, for every discrete time t_j, $j = 0, \dots, N-1$. We denote with $\mathbf{C}_s(j) \in \mathbb{R}^{n_y}$ the vector of measured concentration at iteration j.

As in [11], we assume that the flow dynamic boundary conditions are steady state and for simplicity we suppose that

$$\Gamma_{in} = \bigcup_{l=1}^{n_\theta} \Gamma_{in}^{(l)},$$

being $\Gamma_{in}^{(l)}$ disjoint sets, such that C_{in} is constant on each $\Gamma_{in}^{(l)}$, for all $l = 1, \dots, n_\theta$.

Thus we have to estimate a vector $\boldsymbol{\vartheta}$ of n_θ non negative parameters: equivalently we assume that the function $C_{in} \in H^{\frac{1}{2}}(\Gamma_{in})$ could be identified by a piecewise constant function $C_{in}(\boldsymbol{\vartheta})$ such that

$$C_{in}(\boldsymbol{\vartheta})(\mathbf{x}) = \vartheta(l), \qquad \mathbf{x} \in \Gamma_{in}^{(l)}.$$

Since we are solving an inverse problem we indicate with $\hat{\boldsymbol{\vartheta}}$ the estimate of the real parameters $\boldsymbol{\vartheta}$. Let $\Pi : \mathbb{R}^{N_h} \to \mathbb{R}^{n_y}$ be the map which projects the solution of (4) on the n_y nodes of Γ_{down}: thus we denote with $\Pi(\mathbf{C}(\hat{\boldsymbol{\vartheta}}; t))$ the estimate of $C(C_{in}(\boldsymbol{\vartheta}); t, \mathbf{x})$ at time t on Γ_{down} (*predicted concentration*) obtained solving (1) imposing $C_{in}(\hat{\boldsymbol{\vartheta}})$ on Γ_{in}.

In a space-time discrete setting (9) could be restated as

$$\boldsymbol{\vartheta}^* = \arg \min_{\boldsymbol{\vartheta} \in \mathbb{R}_+^{n_\theta}} \mathcal{F}_d(\boldsymbol{\vartheta}), \tag{10}$$

where the *discrete cost function* is defined as

$$\mathcal{F}_d(\boldsymbol{\vartheta}) := \frac{1}{N}\sum_{j=1}^{N} \left\|\Pi(\mathbf{C}(C_{in}(\boldsymbol{\vartheta}); j)) - \mathbf{C}_s(j)\right\|_2^2 . \tag{11}$$

4.1. Proper Orthogonal Decomposition reduction

Using model order reduction techniques to solve (9), consists in replacing the cost function (11) in (10) with the following one

$$\mathcal{F}_d(\boldsymbol{\vartheta}) := \frac{1}{N}\sum_{j=1}^{N} \left\|\Pi(\tilde{\boldsymbol{C}}(C_{in}(\boldsymbol{\vartheta}); j)) - \mathbf{C}_s(j)\right\|_2^2 \tag{12}$$

where $\tilde{\boldsymbol{C}}$ is the solution of (7). An example of application of POD to solve optimal control problem can be found e.g. in [14].

Since the POD basis depends on the collected snapshots, it is necessary to update the projection space as the estimated control C_{in} varies. Let $\bar{n}$ a small positive integer: at every iteration i in this paper we adopt the following index

$$\mathcal{I}^{(i)} := \frac{1}{\bar{n}} \left\| \sum_{j=1}^{\bar{n}} \tilde{\boldsymbol{C}}(C_{in}(\boldsymbol{\vartheta}^{(i)}); j) - \mathbf{C}(C_{in}(\boldsymbol{\vartheta}^{(i)}); j) \right\|_2^2 ,$$

i.e. we compare the first iterations of the unreduced system with those obtained projecting on the old POD basis used at iteration $i-1$. Only if $\mathcal{I}^{(i)}$ is greater than a fixed threshold, the i-th basis is updated, computing new snapshots, as described in section 2.2. Two strategies can be used [14]: old snapshots can be discarded or not. In practice this consists in adding POD modes computed in the $i-1$-th iteration to the new snapshots ensamble: in this case the projection space is more robust to control variations but usually is slightly bigger. For our experimental tests we prefer to discard old snapshots. We obserse that in [14] a new basis is computed at every iteration, without considering an index $\mathcal{I}$.

5. Known source location $\boldsymbol{\Gamma_{in}}$

As a first step toward the solution strategy, we consider a simpler problem, assuming that the source location Γ_{in} is known.

5.1. Solution uniqueness

In this section we demonstrate that if Γ_{in} is known, then the discrete inverse problem admits a unique solution, since there are no local minima. Moreover changes in C_{in} corresponds to changes in the registered concentration.

First of all we prove the following Lemma, which justifies mathematically the physical principle that, as C_{in} increases on Γ_{in}, the concentration on Γ_{down} increases too.

Lemma 5.1. *Consider the two problems*

$$\left\{\begin{array}{rcllll} \frac{\partial C_i}{\partial t} - \mu\Delta C_i + \nabla\cdot(\boldsymbol{u}C_i) + \sigma C_i & = & 0, & & in & (0,t_f)\times\Omega \\ C_i & = & C_0, & & on & \{0\}\times\Omega \\ C_i & = & C_{in}^{(i)}, & & on & (0,t_f)\times\Gamma_{in} \\ C_i & = & C_{up}, & & on & (0,t_f)\times\Gamma_{up} \\ \mu\frac{\partial C_i}{\partial n} & = & 0, & & on & (0,t_f)\times\Gamma_{down} \\ C_i & = & 0, & & on & (0,t_f)\times\Gamma_r \end{array}\right. \tag{13}$$

represented in Figure 2 (up), where $i = 1,2$. Suppose that $C_{in}^{(2)}(\boldsymbol{x}) > C_{in}^{(1)}(\boldsymbol{x})$, for every $\boldsymbol{x}\in\Gamma_{in}$.

Then $C_2(t,\boldsymbol{x}) > C_1(t,\boldsymbol{x})$ for every $t\in(0,t_f)$ and $\boldsymbol{x}\in\Gamma_{down}$.

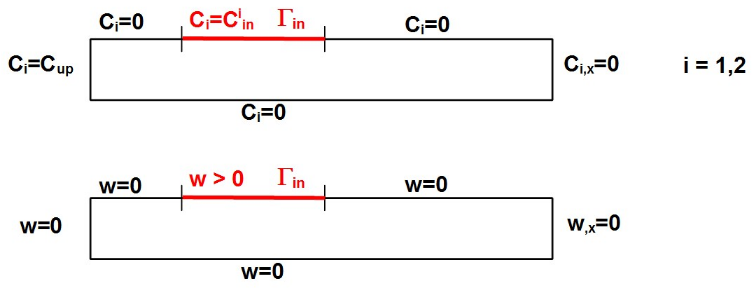

Figure 2:

Proof.

Define $w := C_2 - C_1$, which solves

$$\left\{\begin{array}{rcllll} \frac{\partial w}{\partial t} - \mu\Delta w + \nabla\cdot(\mathbf{u}w) + \sigma w & = & 0, & & in & (0,t_f)\times\Omega \\ w & = & 0, & & on & \{0\}\times\Omega \\ w & = & C_{in}^{(2)} - C_{in}^{(1)}, & & on & (0,t_f)\times\Gamma_{in} \\ w & = & 0, & & on & (0,t_f)\times\Gamma_{up} \\ \mu\frac{\partial w}{\partial n} & = & 0, & & on & (0,t_f)\times\Gamma_{down} \\ w & = & 0, & & on & (0,t_f)\times\Gamma_r \end{array}\right. \tag{14}$$

as illustrated in figure 2 (down).

Observe that w is smooth only inside the domain, but it is not continuous near the boundary, where it admits discontinuities of the first kind: thus generalized solutions must be considered. The strong minimum principle for parabolic operators can be extended for generalized solutions [10, 19] (cfr. appendix **??**): thus we know that the minimum is assumed at the boundary.

Moreover, for every open neighbourhood U of Γ_{down}, such that w is regular inside $U\cap\Omega$, $\frac{\partial w}{\partial n} = 0$ on Γ_{down} implies that the maximum and the minimum of w over $U\cap\Omega$ cannot belong to Γ_{down} (cfr. [10]).

As a consequence $w \geq 0$ in $(0, t_f) \times \Omega$ and, since the minimum is not attained on Γ_{down}, $w = C_2 - C_1 > 0$ on Γ_{down}, for all $t \in (0, t_f)$ i.e. the thesis holds true.

$\square$

The following Proposition is equivalent to prove that there are no local minima.

Proposition 5.1. *For every $\bar{\boldsymbol{\vartheta}} \in \mathbb{R}_+^{n_\theta}$, $\bar{\boldsymbol{\vartheta}} \neq \boldsymbol{\vartheta}^*$, there exists at least a sequence of profiles $\{\boldsymbol{\vartheta}\}_n$, $\boldsymbol{\vartheta}_0 = \bar{\boldsymbol{\vartheta}}$, converging in $\mathcal{L}^2(\mathbb{R}^{n_\theta})$ to the real profile $\boldsymbol{\vartheta}^*$, such that $\mathcal{F}_d(\boldsymbol{\vartheta}_n) \downarrow \mathcal{F}_d(\boldsymbol{\vartheta}^*)$.*

Proof. We can construct the sequence $\{\boldsymbol{\vartheta}_n\}_n$ in the following way. For every $k = 1, \ldots, n_\theta$

$$\vartheta_k(j) := \begin{cases} \vartheta_{k-1}(j), & j \neq k \\ \bar{\vartheta}(j) - (\bar{\vartheta}(j) - \vartheta^*(j)), & j = k \end{cases} . \tag{15}$$

Thus $\boldsymbol{\vartheta}_{n_\theta} = \boldsymbol{\vartheta}^*$ by construction. Moreover the corresponding sequence of cost functions is *decreasing*: $\mathcal{F}_d(\boldsymbol{\vartheta}_1) > \mathcal{F}_d(\boldsymbol{\vartheta}_2) > \ldots > \mathcal{F}_d(\boldsymbol{\vartheta}^*)$. This fact is a direct consequence of the application of Lemma 5.1: suppose that $\vartheta_{k-1}(k) < \vartheta^*(k)$. Then $\vartheta_k(k) > \vartheta_{k-1}(k)$ by construction and thus $\Pi(\mathbf{C}(\boldsymbol{\vartheta}_k; t))$ will be higher than $\Pi(\mathbf{C}(\boldsymbol{\vartheta}_{k-1}; t))$ for every $t \in (0, t_f)$ (Lemma 5.1) and thus closer to $\Pi(\mathbf{C}(\boldsymbol{\vartheta}^*; t))$. Analogously if $\vartheta_{k-1}(k) > \vartheta^*(k)$, applying Lemma 5.1, $\Pi(\mathbf{C}(\boldsymbol{\vartheta}_k; t))$ will be lower than $\Pi(\mathbf{C}(\boldsymbol{\vartheta}_{k-1}; t))$ for all t and thus closer to $\Pi(\mathbf{C}(\boldsymbol{\vartheta}^*; t))$.

$\square$

5.2. Numerical solution strategy

Starting from an initial guess $\hat{\boldsymbol{\vartheta}}^{(0)}$, line search algorithms find the $k+1$-iteration starting from the k-th one in the following way:

$$\hat{\boldsymbol{\vartheta}}^{(k+1)} = \hat{\boldsymbol{\vartheta}}^{(k)} + \alpha^{(k)} \mathbf{s}^{(k)},$$

where the *damping parameter* $\alpha^{(k)}$ is obtained using a bisection procedure.

The standard Newton step consists in solving at each iteration the system

$$\mathcal{F}_d^{''}(\hat{\boldsymbol{\vartheta}}^{(k)}) \mathbf{s}^{(k)} = -\mathcal{F}_d^{'}(\hat{\boldsymbol{\vartheta}}^{(k)}).$$

Let $\mathcal{R} : \mathbb{R}^{n_y \times N} \to \mathbb{R}^{n_y N}$ be the map s.t. starting from an $n_y \times N$ matrix $B = [b_1, \ldots, b_N]$, it gives $\mathcal{R}(B) := \begin{pmatrix} b_1 \\ \vdots \\ b_N \end{pmatrix}$.

The computation of $\mathcal{F}_d^{''}$, Hessian of the cost function, usually is expensive. Moreover, since we are dealing with a *least squares* problem (9), we adopt the Gauss-Newton approximation (cfr. [26]), i.e. we solve

$$\Psi_{\hat{\boldsymbol{\vartheta}}^{(k)}} \mathbf{s}^{(k)} = \mathbf{e}_{\hat{\boldsymbol{\vartheta}}^{(k)}}, \tag{16}$$

where the *sensitivity matrix* $\Psi_{\hat{\boldsymbol{\vartheta}}^{(k)}} \in \mathbb{R}^{n_y N \times n_\theta}$ is such that

$$\Psi_{\hat{\boldsymbol{\vartheta}}^{(k)}}(:,i) := \frac{\partial}{\partial \hat{\boldsymbol{\vartheta}}^{(k)}(i)} \mathcal{R}(\Pi(\mathbf{C}(\hat{\boldsymbol{\vartheta}}^{(k)};\cdot))),$$

for all $i = 1, \dots, n_\theta$ and the *prediction error*

$$\mathbf{e}_{\hat{\boldsymbol{\vartheta}}^{(k)}} := \mathcal{R}(\mathbf{C}_s(\cdot)) - \mathcal{R}(\Pi(\mathbf{C}(\hat{\boldsymbol{\vartheta}}^{(k)};\cdot))).$$

To compute numerically the sensitivity matrix a finite difference scheme is needed:

$$\Psi_{\hat{\boldsymbol{\vartheta}}^{(k)}}(:,j) \approx \frac{1}{\delta}\left[\mathcal{R}(\Pi(\mathbf{C}(\hat{\boldsymbol{\vartheta}}^{(k)}(1),\dots,\hat{\boldsymbol{\vartheta}}^{(k)}(j)+\delta,\dots,\hat{\boldsymbol{\vartheta}}^{(k)}(n_\theta);\cdot))) - \mathcal{R}(\Pi(\mathbf{C}(\hat{\boldsymbol{\vartheta}}^{(k)};\cdot)))\right],$$

where $\delta > 0$ is a small perturbation parameter.

Observe that in general this approximation is computationally expensive, since, it requires the computation of the concentration also for the perturbed input. When Γ_{in} is known, only very few parameters are considered, thus this approximation is effective. The problem becomes more involving when Γ_{in} is unknown, since the number of parameters is higher: in section 6 we will explain how the adaptive parametrization and time localization can reduce the computational cost.

If $\delta > 0$ is too small, the finite difference estimate could be inaccurate, since at the numerator we are considering the difference between two quantities which has approximately the same absolute value, and this is divided by a very small denominator, which amplifies the error. A possible solution e.g. is to adopt the Complex-Step Derivative Approximation [23], in which an immaginary increment $i\delta$ is used, approximating

$$\Psi_{\hat{\boldsymbol{\vartheta}}^{(k)}}(:,j) \approx \frac{1}{\delta} Im\left(\mathcal{R}(\Pi(\mathbf{C}(\hat{\boldsymbol{\vartheta}}^{(k)}(1),\dots,\hat{\boldsymbol{\vartheta}}^{(k)}(j)+i\delta,\dots,\hat{\boldsymbol{\vartheta}}^{(k)}(n_\theta);\cdot)))\right).$$

Finally observe that we are assuming that the pollutant is put into the domain, thus

$$C_{in} \geq 0 :$$

as a consequence we need also a *projection* step onto $[0, +\infty)$ of each component of $\hat{\boldsymbol{\vartheta}}^{(k)}$, after its computation.

5.3. Numerical results

In this section the Projected damped Gauss Newton is compared to other classical solution strategies. Experimental data are simulated numerically, on $\Omega = [0,8] \times [0,1]$, $\Gamma_h = [0,8] \times \{1\} \cup [0,8] \times \{0\}$. Moreover the velocity field $\mathbf{u}$ is modelled as a Poiseuille flow i.e.

$$\mathbf{u}(x_1,x_2) = \begin{pmatrix} -4\nu x_2^2 + 4\nu x_2 \\ 0 \end{pmatrix}.$$

We assume that $\nu = 50$, $\mu = 0.1$, $\sigma = 0.1$ and $C_{up} = 0.1$. Moreover in this section a Gaussian error of variance 0.05 and mean zero is added.

Classical solution strategies cited in this section are well described e.g. in [16]. As a regularization parameter, when needed, we use $\alpha = 0.01$, moreover we choose a maximum number of iterations $max_{it} = 20$. Consider the following two *sparse* examples:

1. $\Gamma_{in} = [4, 4.5] \times \{1\}$, $\vartheta = 100$;

2. $\Gamma_{in} = [4.5, 5] \times \{1\} \cup [1.5, 2] \times \{0\}\,,\ \boldsymbol{\vartheta} = (100, 80)$;

and see how different techniques approximate them.

First of all we consider the example 1. Performances of different methods are depicted in figure 3. In the second example, two parameters have to be estimated: results are plotted in figure 4.

Observe that in both cases the Projected damped Gauss Newton algorithm performs well, converging faster to the optimal solution. It should be noted that, in constrast to Tikhonov and Levenberg-Marquardt it does not need a regularization parameter: this is important because it tells us that, knowing the source location, the inverse problem is not ill-posed, as stated in Proposition 5.1.

5.4. Reduce the order of the system using POD

In this section we analyze the POD reduction introduced in section 2.2 on a test case. Consider example 2 introduced in the previous section; in POD reduction two parameters plays a central role: t_m, which characterizes the interval $[t_0, t_m]$ when snapshots are collected, and the threshold τ_σ on the singular values of the snapshots matrix. As can be seen in table 1, increasing t_m corresponds to a better approximation, since more snapshots are collected. To obtain higher accuracy decreasing t_m, it is necessary to increase τ_σ, i.e. to consider a bigger reduced model.

		L^1 error:		$\mathcal{F}_d(\boldsymbol{\vartheta})$	Dim. model	num. it.
		up	down			
Unreduce model		0	0	10^{-20}	1071	2
Reduced models:						
t_m	τ_σ					
2.5	0.01	0.117	4.8	0.113	23	5
2.5	10^{-4}	0.083	1.24	0.089	32	4
3.75	0.01	0.08	0.08	$6 \cdot 10^{-4}$	25	3
3.75	10^{-4}	0.02	0.02	$3 \cdot 10^{-5}$	39	4
5	0.01	0.0015	0.0015	10^{-6}	29	3

Table 1: *Example 2 of section 5.3, choosing different intervals $[t_0, t_m]$ to collect snapshots and different thresholds τ_σ on singular values of the snapshots matrix.*

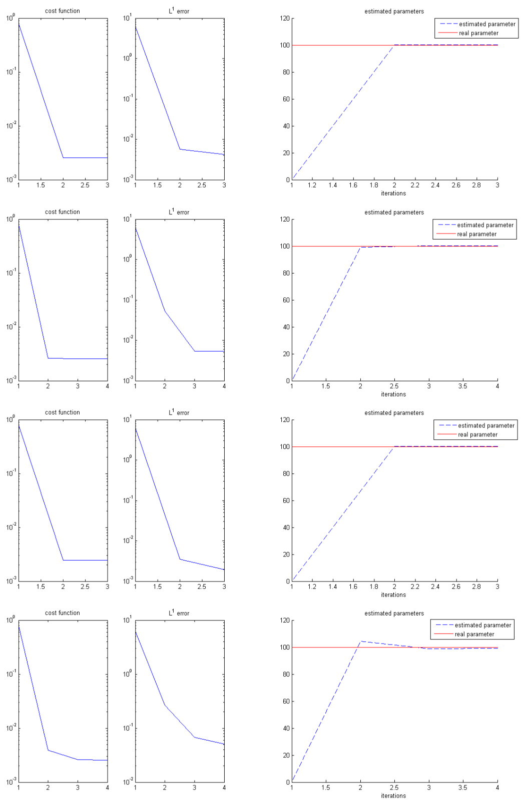

Figure 3: *First example: different strategies. Left: cost function and error, right: convergence. First row: projected damped Gauss Newton, second row: Levenberg Marquardt, third row: steepest descent, fourth row: Tikhonov method.*

It is important to note that the reduction is significative with respect to the unreduced model, which has dimension 1071. However, as described in section 2.2, it should be noted that it is necessary to update the POD basis: in all these examples the basis is updated at every new iteration, imposing 0.1 as a threshold on $\mathcal{I}^{(i)}$.

A more involving problem is considered in section 6, where it is assumed that also the source location Γ_{in} is unknown. In general in that case projected damped Gauss Newton could not be sufficient and it is too costly, thus it is necessary to adopt a suitable solution strategy based upon an adaptive parametrization and time localization.

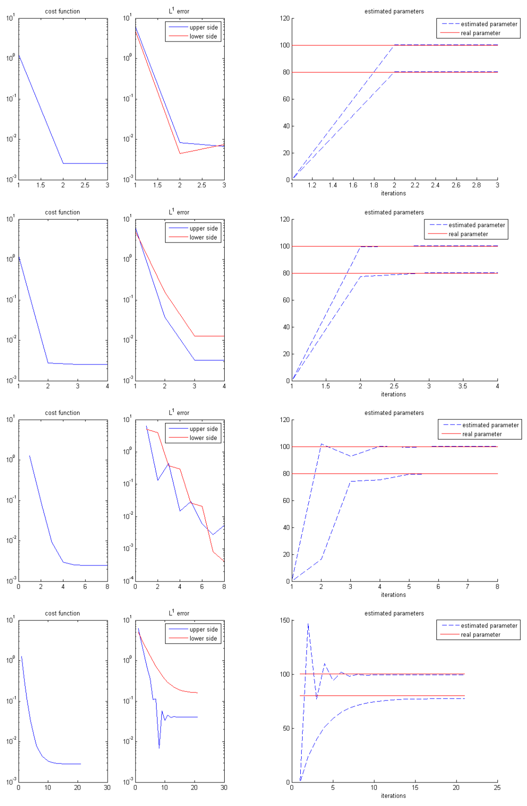

Figure 4: *Second example: different strategies. Left: cost function and error, right: convergence. First row: projected damped Gauss Newton, second row: Levenberg Marquardt, third row: steepest descent, fourth row: Tikhonov method.*

6. Unknown source location Γ_{in}

Suppose now that the *location* of Γ_{in} is unknown.

6.1. Ill-posedness of the problem

To study analytically what happens when Γ_{in} is unknown, we consider a simplified model problem: let $C = C(x)$, $x \in [x_1, x_2] \subset \mathbb{R}$, $x_2 > x_1$ be the solution of the following one dimensional ODE:

$$\left\{ \begin{array}{rcll} -\mu C^{''}(x) + uC^{'}(x) & = & f(x), & \quad in\ (x_1, x_2), \\ C(x_1) & = & C_{up}, & \\ C^{'}(x_2) & = & 0, & \end{array} \right. \tag{17}$$

where $f(x) = \left\{ \begin{array}{ll} M, & |x - x_m| \le h \\ 0, & elsewhere\ in\ (x_1, x_2) \end{array} \right.$, $M > 0$, $x_m \in (x_1, x_2)$, $h \in (0, 1)$ s.t. $x_m \pm h \in (x_1, x_2)$. Observe that (17) can be viewed as the one dimensional stationary counterpart of (1) when $\sigma = 0$ and considering only the x-axis in figure 1: the unknown immision boundary Γ_{in} can be represented by an unknown forcing term f, applied in $[x_m - h, x_m + h]$, of intensity M. In this context the inverse problem (9) is equivalent to determine the source position (h and x_m) and intensity (M) given the measured concentration $C_s \in \mathbb{R}$ in x_2.

Problem (17) can be solved analytically, obtaining

$$C(x) = \begin{cases} c_1 + c_2 e^{\frac{u}{\mu}x}, & x < x_m - h, \\ c_3 + \frac{M}{u}x + c_4 e^{\frac{u}{\mu}x}, & |x - x_m| \leq h, \\ c_5 + c_6 e^{\frac{u}{\mu}x}, & x > x_m + h, \end{cases}$$

where $c_1, \ldots, c_6$ are suitable real coefficients obtained imposing boundary conditions and continuity of u and u' in $x_m \pm h$. In particular we are interested in estimating the concentration at the measurement point $x = x_2$. For simplicity we assume that $x_1 = 0$ and $x_2 = 1$. Thus it can be derived that

$$c_6 = 0, \qquad c_5 = \frac{1}{u^2} \exp \frac{-u(x_m + h)}{\mu} \left(2uhM \exp \frac{u(x_m + h)}{\mu} + \mu M \left(1 - \exp \frac{2uh}{\mu} \right) \right).$$

Thus $C(x)$ is constantly equal to c_5 in $[x_m + h, 1]$. We can now study how

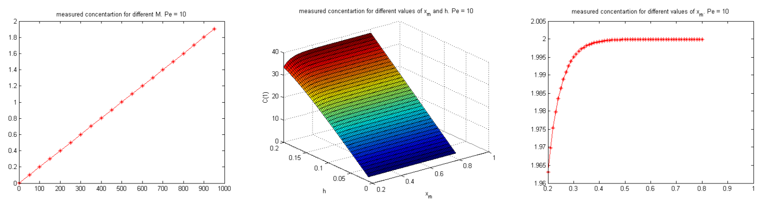

Figure 5: *Solution of (17) at the measurement point $x_2 = 1$ for different values of M (left), h and x_m (center), x_m (right).*

$C(1)$ depends on M, h, L. We consider $\mu = 0.5$ and $u = 10$ (Peclet number $Pe = \frac{u}{2\mu} = 10$, quantity that characterize convection diffusion problems). As can be seen in figure 5, varying only M, fixing h and x_m (i.e. knowing the source location), corresponds to a linear striclty increasing $C(1)$ (cfr. figure 5 (left)). On the contrary fixing M but varying h and x_m corresponds to the surface plotted in figure 5 (center): fixing h for different values of x_m we obtain almost the same $C(1)$ (cfr. figure 5 (right)). Thus measuring $C(1)$, *the problem of determining the source is ill-posed in the stationary regime.* Increasing the Peclet number this phenomenum is stressed.

Even for this simplified 1D stationary problem, in general unknown source position gives rise to an ill-conditioned problem.

6.2. Numerical solution of the discrete inverse problem

The problem consists in estimating both *the position* of the sources C_{in} in the horizontal segments $\Gamma_h := \Gamma_r \cup \Gamma_{in}$ and their *intensity.*

6.2.1. Algorithm 1: working on the finest subdivision

First of all we consider $\left\{x_1, \ldots, x_{\frac{n_\theta^{(f)}}{2}+1}\right\}$ a *reference uniform finest subdivision* of Γ_h of step length Δx, which represents the minimum amplitude of estimated source emissions.

The simplest strategy consists in applying the Gauss Newton method directly on the finest subdivision (cfr. algorithm 1), i.e. in estimating $n_\theta^{(f)}$ parameters. This problem is particularly demanding for its high computational cost, due to the large number of parameters to be estimated at each Newton's iteration. Moreover it should be noted that if we want to estimate a *sparse* vector of parameters, working only on the finest subdivision is not efficient, as we will see in the following sections. To solve the system (16) both *TSVD* and

Algorithm 1 Sketch of the algorithm working on the finest subdivision:

```
1: Given the finest subdivision of $\Gamma_h$, $\hat{\boldsymbol{\theta}}^0 = \mathbf{0}$, $\mu^0 = 1$;
2: while $\mathcal{F}_d(\hat{\boldsymbol{\vartheta}}^l) < tol$ do
3:     solve $\psi_{\hat{\theta}^k}\mathbf{s}^k = \mathbf{e}_{\hat{\theta}^k}$ ;
4:     $\hat{\boldsymbol{\theta}}^{k+1} = \hat{\boldsymbol{\theta}}^k + \mu^k \mathbf{s}^k$
5:     projection: for every $j \in [0, n_\theta - 1]$ s.t. $\hat{\theta}^{k+1}(j) < 0$, impose $\hat{\theta}^{k+1}(j) = 0$
6:     compute $\mathcal{F}_n(\hat{\boldsymbol{\theta}}^{k+1})$
7:     if $\mathcal{F}_n(\hat{\boldsymbol{\theta}}^{k+1}) > \mathcal{F}_n(\hat{\boldsymbol{\theta}}^k)$ then
8:         $l = 0$;
9:         $\mu^{k,l} = \frac{\mu^k}{2}$
10:        while $\mathcal{F}_n(\hat{\boldsymbol{\theta}}^{k+1}) < \mathcal{F}_n(\hat{\boldsymbol{\theta}}^k)$ do
11:            $\hat{\boldsymbol{\theta}}^{k+1} = \hat{\boldsymbol{\theta}}^k + \mu^{k,l}\mathbf{s}^k$
12:            $l = l + 1$;
13:            $\mu^{k,l} = \frac{\mu^{k,l}}{2}$
14:        end while
15:    end if
16: end while
```

diagonal scaling [26] are used. The last one, presented in [21] to solve a conduction inverse problem, works as follows: at iteration k, given the subdivision $\mathcal{S}^{(k)} = \left\{x_1, \ldots, x_{\frac{n_\theta^{(f)}}{2}+1}\right\}$, for every $i = 1, \ldots, n_c^{(k)}$, where $n_c^{(k)} = n_\theta^{(f)}$ denotes the number of columns of $\Psi_{\hat{\boldsymbol{\vartheta}}^{(k)}}$ at iteration k, $\Psi_{\hat{\boldsymbol{\vartheta}}^{(k)}}(:, i)$ is multiplied by a weight d_i, equal to the length of the maximal segment of the current subdivision, divided by the length of the segment corresponding to the i-th column. Thus diagonal scaling corresponds to solve

$$\begin{aligned} \Psi_{\hat{\boldsymbol{\vartheta}}^{(k)}} D^{(k)} \tilde{\mathbf{s}}^{(k)} &= \mathbf{e}_{\hat{\boldsymbol{\vartheta}}^{(k)}}, \qquad D^{(k)} = diag(d_i^{(k)}), \qquad d_i^{(k)} = \frac{\max_{x_{j+1}^k, x_j^k \in \mathcal{S}^{(k)}} x_{j+1}^k - x_j^k}{x_{i+1}^k - x_i^k}, \\ \mathbf{s}^{(k)} &= D^{(k)} \tilde{\mathbf{s}}^{(k)}, \end{aligned} \tag{18}$$

instead of (16).

6.2.2. Algorithm 2: working on the finest subdivision with time localization

As explained in section 6.1, in the stationary regime the problem is illposed: time localization corresponds to a better conditioned problem, since it consists in selecting only those rows of the sensitivity matrix which are significative for the dynamic, i.e. corresponding to the transitional dynamics.

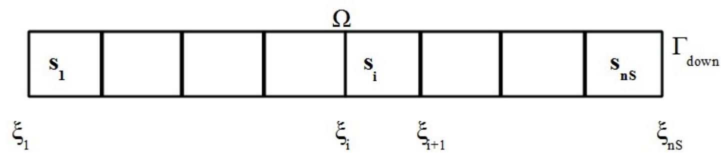

Figure 6: *Example of partition of* Ω *in sections.*

More precisely, the idea is to partition the domain Ω in a suitable number $n_s > 1$ of *sections* $\mathcal{U} = \{s_j\}$, $j = 1, \ldots, n_s$ (cfr. e.g. figure 6). Referring to figure 1, we suppose that $s_j := [\xi_j, \xi_{j+1}] \times [y_1, y_2]$, $\xi_1 = x_1$, $\xi_{n_s+1} = x_2$. In particular in algorithm 2 we assume that $\{\xi_1, \ldots, \xi_{n_s+1}\} = \left\{x_1, \ldots, x_{\frac{n_\theta^{(f)}}{2}+1}\right\}$, i.e. it coincides with the finest subdivision. Denote with $I^{(j)}$, the set of parameters belonging to s_j.

The algorithm works as follows: starting from s_{n_s}, it computes the sensitivity matrix only of those parameters belonging to $I^{(n_s)}$, only in the time interval $[t_0^{(n_s)}, t_f^{(n_s)}]$, $t_0^{(n_s)} \geq t_0$, $t_f^{(n_s)} \leq t_f$: below it is explained how to choose the interval. The estimate of the parameters of section n_s is done as explained before, using a projected damped Gauss-Newton method. To regularize the problem both TSVD and diagonal scaling are used.

Define $\mathcal{O}^{(j)}$, $j = 1, \ldots, n_s - 1$, the set of parameters estimated in section s_{j+1} greater than a threshold $\epsilon_3 > 0$: then in section s_j all parameters belonging to $\mathcal{O}^{(j)} \cup I^{(j)}$ will be estimated, only in the time interval $[t_0^{(j)}, t_f^{(j)}]$. In algorithm 2 previous ideas are summarized.

6.2.3. Algorithm 3: using an adaptive parametrization

The alternative is to use an *adaptive parametrization*, i.e. to adaptively update the subdivision of Γ_h used in the current iteration of the Newton method. This strategy is particularly indicated when dealing with a *sparse* vector of parameters: in this situation it is important to localize Γ_{in} in Γ_h and to refine the parametrization possibly only around that point. It reduces the computational cost reducing the number of columns of the sensitivity matrix. A similar strategy has been presented in [8], to solve an inverse conduction problem of corrosion estimation.

The algorithm works as follows: starting from an initial *coarse* subdivision of Γ_h, $\mathcal{S}^{(1)}$, at the k-th iteration the algorithm first computes a Gauss-Newton

Algorithm 2 Sketch of the algorithm working on the finest subdivision with time localization:

```
1: Given $\{\xi_1, \dots, \xi_{n_s+1}\}$ coincident with the finest subdivision of $\Gamma_h$ and the threshold $\epsilon_3 > 0$;
2: while $\mathcal{F}_d(\hat{\boldsymbol{\vartheta}}^l) < tol$ do
3:     $i = n_s$; $\mathcal{O}^{(n_s)} = \emptyset$
4:     while $i > 0$ do
5:         Let $I^{(i)}$ be the set of parameters of $\hat{\boldsymbol{\vartheta}}^l$ that belongs to section $i$;
6:         in $[t_0^{(i)}, t_f^{(i)}]$ apply the regularized projected damped Gauss Newton method to optimize parameters whose indices belong to $I^{(i)} \cup \mathcal{O}^{(i)}$;
7:         update the positions $I^{(i)} \cup \mathcal{O}^{(i)}$ of $\hat{\boldsymbol{\vartheta}}^l$;
8:         define $\mathcal{O}^{(i-1)}$ as the set of indices of parameters greater than $\epsilon_3$;
9:         $i = i - 1$;
10:    end while
11: end while
```

iteration $\hat{\boldsymbol{\vartheta}}^{(k)} \in \mathbb{R}^{n_\theta}$. For every element of $\hat{\boldsymbol{\vartheta}}^{(k)} \in \mathbb{R}^{n_\theta}$ greater than a fix threshold $\epsilon_1 > 0$, the segment of $\mathcal{S}^{(k)}$ corresponding to that parameter is bisected: thus a new subdivision $\mathcal{S}^{(k+1)}$ is defined adding to $\mathcal{S}^{(k)}$ all the computed middle points. Finally are selected only those parameters which are greater than a fixed threshold $\epsilon_2 > 0$, and we indicate with $\Lambda^{(k)}$ this ensemble; the other parameters remain constant in the following iteration. The main ideas of the adaptive algorithm are sketched in algorithm 3.

To avoid a large over-refinement, the bisection procedure can be limited, for example applying it at each iteration only a certain number of times, choosing the segments to be refined as those corresponding to greater parameters.

6.2.4. Algorithm 4: using an adaptive parametrization and time localization

The idea now is to reduce both the number of columns and of rows of the sensitivity matrix. As in algorithm 2, the domain Ω is partitioned in $n_s > 1$ *sections* $\mathcal{U} = \{s_j\}$, $j = 1, \dots, n_s$, however in algorithm 4 we assume that $\{\xi_1, \dots, \xi_{n_s+1}\} = \mathcal{S}^{(1)}$, i.e. it coincides with the coarse initial subdivision applied in the adaptive strategy. In section s_j, considering the time interval $[t_0^{(j)}, t_f^{(j)}]$, all parameters belonging to $\mathcal{O}^{(j)} \cup I^{(j)}$ will be estimated, and the adaptive procedure will be applied until a minimum is reached. Observe that this coincides with an internal loop: the ideas are summarized in algorithm 4.

6.2.5. Time localization: how to choose time intervals $[t_0^{(i)}, t_f^{(i)}]$

A key point is the choice of the local time intervals $[t_0^{(i)}, t_f^{(i)}]$, for every section s_i, $i = 1, \dots, n_s$, $t_0^{(i)} \geq t_0$ and $t_f^{(i)} \leq t_f$. The i-th interval must be chosen such that it contains the transitional dynamics of section s_i but not that of sections s_j, $j < i$. To describe more clearly this idea, consider the model problem introduced in section 5.3: moreover suppose for simplicity that $n_s = 2$, $\{\xi_1, \xi_2, \xi_3\} = \{0, 4, 8\}$, as depicted in figure 7, and consider as the finest subdivision a uniform one of step length 0.5. Consider figure 8: the j-th

Algorithm 3 Sketch of the adaptive algorithm:

```
1: Given the finest subdivision of $\Gamma_h$ of step length $\Delta x$, the tolerance $tol > 0$ and thresholds $\epsilon_1, \epsilon_2 > 0$, consider the coarse subdivision $\mathcal{S}^{(1)} = \left\{ x_1^1, \dots, x^1_{\frac{n_\theta^1}{2}+1} \right\}$, of $\Gamma_h$;
2: $\hat{\boldsymbol{\vartheta}}^1 = \mathbf{0}_{n_\theta^1} \in \mathbb{R}^{n_\theta^1}$;
3: $l = 1$, $\Lambda^{(1)} = [1, \dots, n_\theta^1]$, set of indexes of parameters to be optimized
4: while $\mathcal{F}_d(\hat{\boldsymbol{\vartheta}}^l) < tol$ do
5:     $\mathcal{S}^{(l+1)} := \left\{ x_1^{l+1}, \dots, x^{l+1}_{\frac{n_\theta^{l+1}}{2}+1} \right\} = \mathcal{S}^{(l)}$;
6:     $n_\theta^{l+1} = n_\theta^l$, $I = n_\theta^{l+1}$
7:     for all $i \in [1, I]$ do
8:         if $\hat{\theta}^l(i) > \epsilon_1$% bisect the corresponding segment then
9:             $n_\theta^{l+1} = n_\theta^{l+1} + 1$, $I = I + 1$;
10:            let $[x^{l+1}(\hat{\theta}^l(i)), x^{l+1}(\hat{\theta}^l(i))]$ be the segment corresponding to parameter $\hat{\theta}^l(i)$;
11:            $\mathcal{S}^{(l+1)} = \mathcal{S}^{(l+1)} \cup \frac{x^{l+1}(\hat{\theta}^l(i)) - x^{l+1}(\hat{\theta}^l(i))}{2}$,
12:        end if
13:    end for
14:    given the subdivision $\mathcal{S}^{(l+1)}$ apply the projected damped Gauss-Newton method, optimizing only parameters whose indexes belong to $\Lambda^{(k)}$, obtaining $\hat{\boldsymbol{\vartheta}}^{l+1} \in \mathbb{R}^{n_\theta^{l+1}}$
15:    $\Lambda^{(k)} = \emptyset$;
16:    for all $i \in [1, I]$ do
17:        if $\hat{\vartheta}^{l+1}(i) > \epsilon_2$ then
18:            $\Lambda^{(k)} = \Lambda^{(k)} \cup i$;
19:        end if
20:    end for
21:    $l = l + 1$;
22: end while
```

Algorithm 4 Sketch of the adaptive algorithm with time localization:

```
1: Given the partition of $\Omega$ $\{\xi_1, \dots, \xi_{n_s+1}\} = \mathcal{S}^{(1)}$, the thresholds $\epsilon_1, \epsilon_2, \epsilon_3 > 0$, $\hat{\boldsymbol{\vartheta}}^0 = \mathbf{0}$, $j = 0$;
2: while $\mathcal{F}_d(\hat{\boldsymbol{\vartheta}}^j) < tol$ do
3:     $j = j + 1$;
4:     $i = n_s$; $\mathcal{O}^{(n_s,1)} = \emptyset$
5:     while $i > 0$ do
6:         $l = 1$, $\Lambda^{(i,1)} = [1, \dots, n_\theta^{j,i,1}]$, set of indexes of parameters to be optimized
7:         while a minimum is reached do
8:             Let $I^{(i,l)}$ be the set of parameters of $\hat{\boldsymbol{\vartheta}}^{j,i,l}$ that belongs to section $i$;
9:             in $[t_0^{(i)}, t_f^{(i)}]$ apply the regularized projected damped Gauss Newton method to optimize parameters whose indices belong to $I^{(i,l)} \cup \mathcal{O}^{(i,l)}$;
10:            update the positions $I^{(i,l)} \cup \mathcal{O}^{(i,l)}$ of $\hat{\boldsymbol{\vartheta}}^{j,i,l}$;
11:            apply the adaptive strategy:
12:            $\mathcal{S}^{(j,i,l+1)} = \mathcal{S}^{(j,i,l)}$;
13:            $n_\theta^{j,i,l+1} = n_\theta^{j,i,l}$, $I = n_\theta^{j,i,l+1}$
14:            for all $k \in [1, I]$ do
15:                if $\hat{\theta}^{j,i,l}(k) > \epsilon_1$ then
16:                    update $\mathcal{S}^{(j,i,l+1)}$, bisecting the segment corresponding to $\hat{\theta}^{j,i,l}(k)$;
17:                end if
18:            end for
19:            given the subdivision $\mathcal{S}^{(j,i,l+1)}$ apply the projected damped Gauss-Newton method, optimizing only parameters whose indexes belong to $\Lambda^{(j,i,l+1)}$
20:            $\Lambda^{(j,i,l+1)} = \emptyset$;
21:            for all $k \in [1, I]$ do
22:                if $\hat{\vartheta}^{j,i,l+1}(k) > \epsilon_2$ then
23:                    $\Lambda^{(j,i,l+1)} = \Lambda^{(j,i,l+1)} \cup k$;
24:                end if
25:            end for
26:            $l = l + 1$;
27:            if the subdivision has been refined, update $\mathcal{O}^{(i,l)}$
28:        end while
29:        $\mathcal{S}^{(j,i)} = \mathcal{S}^{(j,i,l)}$;
30:        define $\mathcal{O}^{(j,i-1)}$ as the set of indices of parameters greater than $\epsilon_3$;
31:        $i = i - 1$;
32:    end while
33:    $\hat{\boldsymbol{\vartheta}}^j = \hat{\boldsymbol{\vartheta}}^{j,i,l}$
34: end while
```

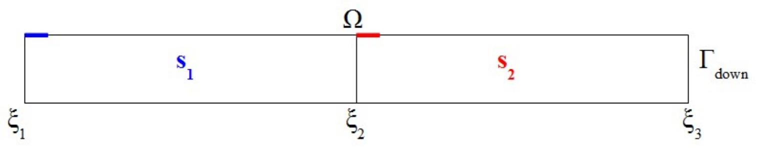

Figure 7: *Partition of* Ω *in* 2 *sections to draw the curves of figure 8: to obtain the red (blue) curve of figure 8, it is considered the mean concentration on* Γ_{down}*, obtained imposing a control different from zero only in the most left segment of the finest subdivision of the upper horizontal segment of section* s_2 *(*s_1*), indicated in red (blue).*

curve ζ_j, $j = 1, 2$, represents the mean concentration (left) and its derivative (right) at the outflow when the boundary control is different from zero only in the most left position of s_j with respect to the finest subdivision. The interval

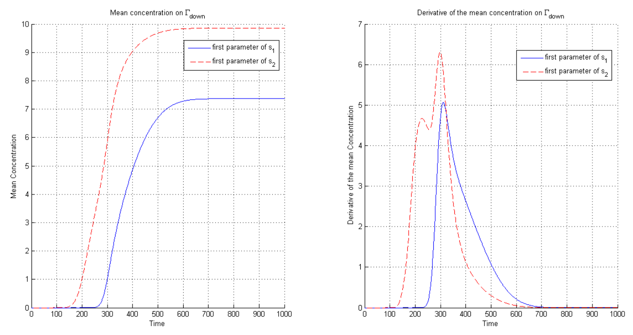

Figure 8: *Time evolution of the mean concentrations at* Γ_{down}*,* ζ_1 *and* ζ_2*, (left) and their derivative (right) for different boundary controls: the boundary control is different from zero only in the most left position of the finest subdivision of* s_1 *(blue) and* s_2 *(red).*

corresponding to s_2 can be $[t_0^{(2)}, t_f^{(2)}] = [180, 260]$, when the red dotted curve corresponding to s_2, ζ_2 , is increasing (transitional regime) and the blue curve corresponding to s_1, ζ_1, is flat, i.e. when only the pollutant put into Ω in s_2 could reach Γ_{down}. While in s_1 the choice can be $[t_0^{(1)}, t_f^{(1)}] = [240, 400]$, since in this interval the transitional regime of s_1 occurs, as showed by ζ_1. This intervals are used in section 6.4, to test algorithm 4.

The previous idea can be extended more rigorously to a general number of sections: let ζ_i, $i = 1, \ldots, n_s$, be the mean concentration at the outflow Γ_{down} when the boundary control is different from zero only in the most left position of s_i, with respect to the finest subdivision. Consider a small threshold $\epsilon_4 > 0$, and two positive parameters $d, D > 0$. Given

$$\begin{aligned} t_0^{(n_s)} &= \min_{t\in[t_0,t_f]} \left\{ \zeta'_{n_s}(t) > \epsilon_4 \text{ and } \zeta'_{n_s-1}(t) < \epsilon_4 \right\}, \\ t_f^{(n_s)} &= \max_{t\in[t_0,t_f]} \left\{ \zeta'_{n_s}(t) > \epsilon_4 \text{ and } \zeta'_{n_s-1}(t) < \epsilon_4 \right\}, \end{aligned}$$

then for $i = 1, \ldots, n_s - 1$

$$\begin{aligned} t_0^{(i)} &= t_f^{(i+1)} - d, \\ t_f^{(i)} &= \begin{cases} \max_{t\in[t_0,t_f]} \left\{ \zeta'_i(t) > \epsilon_4 \text{ and } \zeta'_{i-1}(t) < \epsilon_4 \right\}, & i > 1 \\ \min \left\{ t_f^{i+1} + D, \max_{t\in[t_0,t_f]} \left\{ \zeta'_i(t) > \epsilon_4 \text{ and } \zeta'_{i-1}(t) < \epsilon_4 \right\} \right\}, & i = 1. \end{cases} \end{aligned}$$

The parameter d allows a small overlapping between local time intervals, while D could limit the length of the inteval $[t_0^{(1)}, t_f^{(1)}]$: e.g. in the example presented above, considering $n_s = 2$, $d = 0.2$ and $D = 1.6$.

Observe that the definition of the intervals $[t_0^{(i)}, t_f^{(i)}]$ depends on the shape of the domain, on the velocity field and on the coefficients of the PDE (1): each time one of them is changed, also the intervals should be estimated, observing the transitional dynamics of each section, as explained above.

6.3. Comparing computational costs

In this section we compare the computational costs of the four algorithms.

The first one consists in using the finest subdivision, with the projected damped Gauss Newton strategy. The computational cost of each iteration is pretty high: since the solution of the direct problem has cost NN_h^3, computing the sensitivity matrix $\Psi_{\boldsymbol{\vartheta}} \in \mathbb{R}^{n_y N \times n_\theta^{(f)}}$ has cost $n_\theta^{(f)} N N_h^3$, where $n_\theta^{(f)}$ is the number of parameter of the finest subdivision, which is maximal. Moreover computing the SVD to obtain the new iteration has cost $4n_y^2 N^2 n_\theta^{(f)} + 8Nn_y(n_\theta^{(f)})^2 + 9(n_\theta^{(f)})^3$. Finally computing the new prediction error has cost NN_h^3.

To decrease the cost, the idea is to consider a sensitivity matrix of lower dimensions. The second algorithm consists in combining the finest subdivision with localization in time. The number of sections in this case coincides with one half of the number of parameters of the finest subdivision $n_\theta^{(f)}$. At each iteration k, for every section $i = 1, \ldots, n_s$, $n_s = \frac{n_\theta^{(f)}}{2}$, computing $\Psi_{\boldsymbol{\vartheta}}^{(i)} \in \mathbb{R}^{n_y \frac{t_f^{(i)} - t_0^{(i)}}{Dt} \times n_\theta^{(k,i)}}$ costs $n_\theta^{(k,i)} (\frac{t_f^{(i)} - t_0^{(i)}}{Dt}) N_h^3$, where $n_\theta^{(k,i)}$ denotes the cardinality of $I^{(i)} \cup \mathcal{O}^{(i)}$. Moreover computing the SVD to obtain the new iteration has cost $4n_y^2 \left(\frac{t_f^{(i)} - t_0^{(i)}}{Dt} \right)^2 n_\theta^{(k,i)} + 8 \frac{t_f^{(i)} - t_0^{(i)}}{Dt} n_y (n_\theta^{(k,i)})^2 + (n_\theta^{(k,i)})^3$. Finally computing the new prediction error has cost NN_h^3. Although an higher number of systems must be solved, the algorithm is less costly since the sensitivity matrix has much lower dimensions.

Another possibility to decrease the cost of algorithm one, is to use the third algorithm, which consists in adopting an adaptive parametrization. At the $k-$th iteration computing the sensitivity matrix $\Psi_{\boldsymbol{\vartheta}} \in \mathbb{R}^{n_y N \times n_\theta^{(k)}}$ has cost $n_\theta^{(k)} N N_h^3$, where the number of parameter $n_\theta^{(k)}$ varies during the iterations and $n_\theta^{(k)} < n_\theta^{(f)}$. Moreover computing the SVD to obtain the new iteration has cost $4n_y^2 N^2 n_\theta^{(k)} + 8Nn_y(n_\theta^{(k)})^2 + 9(n_\theta^{(k)})^3$. Finally computing the new prediction error has cost NN_h^3. The gain with respect to the first strategy is evident if $n_\theta^{(k)} << n_\theta^{(f)}$.

The fourth algorithm combines both time localization and the adaptive parametrization. The number of sections in this case coincides with one half the number of parameters of the initial coarse subdivision $\mathcal{S}^{(1)}$. The difference with respect to the second algorithm is that the number of sections n_s is lower, because it is no more related to the finest subdivision: in fact the adaptive parametrization guides the choice of parameters to be estimated at each iteration. However the introduction of the adaptive parametrization introduces an inner loop. In detail, at each iteration k, for every section $i = 1, \dots, n_s$, applying the adaptive procedure until a minimum is reached (index l), computing $\Psi_{\boldsymbol{\vartheta}}^{(k,i,l)} \in \mathbb{R}^{n_y \frac{t_f^{(i)} - t_0^{(i)}}{Dt} \times n_\theta^{(k,i,l)}}$ costs $n_\theta^{(k,i,l)} (\frac{t_f^{(i)} - t_0^{(i)}}{Dt}) N_h^3$. Moreover computing the SVD to obtain the new iteration has cost $4n_y^2 \left(\frac{t_f^{(i)} - t_0^{(i)}}{Dt} \right)^2 n_\theta^{(k,i,l)} + 8 \frac{t_f^{(i)} - t_0^{(i)}}{Dt} n_y (n_\theta^{(k,i,l)})^2 + (n_\theta^{(k,i,l)})^3$. Finally computing the new prediction error has cost NN_h^3.

Just to give an idea of the computational gain of the fourth algorithm, computational costs of the four algorithms are summarized in table 2, averaging results of tests presented in section 6.4.

	Finest subdivision	**Finest subdivision + time localization**	**Adaptive subdivision**	**Adaptive subdivision + time localization**
Computational cost	$8 \cdot 10^{14}$	$2 \cdot 10^{13}$	$5 \cdot 10^{12}$	$8 \cdot 10^{11}$

Table 2: *Estimated computational cost of the four algorithms: using the finest subdivision and the projected damped Gauss Newton method, using the finest subdivision and the localization in time, using the adaptive parametrization and using the adaptive parametrization and time localization.*

6.4. Numerical results

In this section we present some numerical tests to verify the effectiveness of the algorithm. As in section 5.3, experimental data are simulated numerically, on $\Omega = [0, 8] \times [0, 1]$, $\Gamma_h = [0, 8] \times \{1\} \cup [0, 8] \times \{0\}$. Moreover the velocity field $\mathbf{u}$ is modeled as a Poiseuille flow i.e.

$$\mathbf{u}(x_1, x_2) = \begin{pmatrix} -4\nu x_2^2 + 4\nu x_2 \\ 0 \end{pmatrix}.$$

We assume that $\mu = 0.1$, $\sigma = 0.1$ and $C_{up} = 0.1$. Moreover we consider the finest subdivision with step length $\Delta x = 0.5$. In algorithm 2 we consider $\{\xi_1, \ldots, \xi_{n_s+1}\}$ coincident with the finest subdivision, while in algorithm 4 $n_s = 2$ and $\{\xi_1, \ldots, \xi_{n_s+1}\} = \{0, 4, 8\}$. Define *optimal subdivision* the one which describes the real profile with the minimum number of parameters using the bisection criterium. With *distance from the optimal subdivision* we indicate the number of points added (sign +) or subtracted (sign -) to the optimal sundivision. We consider 9 test cases: results for the adaptive strategy with localization in time are shown in figure 9. In table 3, four algorithms are compared: using the finest subdivision and the projected damped Gauss Newton method, using the finest subdivision and the localization in time, using the adaptive parametrization and using the adaptive parametrization and time localization.

First of all observe that the number of iterations of algorithms 2 and 4 is higher since also sub-iterations to reach the minimum inside each section are counted (inner loop).

In tests 1, 2 and 7, also working on the finest subdivision performs well, but it is much more costly. When the condition number of the sensitivity matrix $\Psi_{\boldsymbol{\vartheta}}$ increases, the accuracy is low. In particular in tests 3, 4, 5 it is evident how time localization improves convergence results both in algorithms 2 and 4, selecting only some rows of $\Psi_{\boldsymbol{\vartheta}}$. However adopting only time localization is not sufficient in tests 6,7,9. Using an adaptive parametrization corresponds to select only some columns of $\Psi_{\boldsymbol{\vartheta}}$, considering a less number of parameters: in algorithm 3 the number of points added to the optimal subdivision is very low, but in general the estimates tend to be too much approximated. The best strategy consists in combining both adaptive parametrization and time localization (algorithm 4): this is a good compromise between good estimates and reasonable computational cost. Its effectiveness is evident e.g. in tests 8 and 9. Moreover it only adds few points to the optimal subdivision.

In figure 10 and 11 different iterations of the algorithm are shown for test 8: it is evident how the algorithm firstly optimize parameters of section $s_2 = [4, 8] \times [0, 1]$ (figure 11), and then that of $s_1 = [0, 4] \times [0, 1]$ (in figure 10 the first 7 iteration are identical to the first one, thus only iterations 1 and 8 are plotted). The estimated subdivision is sketched in figure 12: it is evident how algorithm 4 slightly over-refine the optimal subdivision.

6.5. Conditioning of the problem

The ill-conditioning of the system matrix $\Psi_{\hat{\boldsymbol{\vartheta}}}$ could increase when smaller segments are considered in Γ_h: in fact in this case consecutive columns tend to be close to linear dependence, due to the small distance (Δx) of the corresponding nodes in Γ_h. This can be demonstrated numerically: consider in fact example 2 presented in section 5.3 and generalize it considering the following parametric problem

$$\Gamma_{in} = [5 - h, 5] \times \{1\} \cup [2 - h, 2] \times \{0\}, \ \boldsymbol{\vartheta} = (100, 80), \qquad 0 < h \leq 2.$$

Even supposing to know source localition Γ_{in}, solving the problem for different values of $h = \{0.0625, 0.125, 0.25, 0.5, 1, 2\}$ and computing the condition number of the sensitivity matrix, it can be seen that as h decreases, the condition number increases (cfr. figure 13). Since the condition number of the sensitivity matrix could become higher when smaller segments are considered, working on the finest subdivision could not be effective to reduce the ill-conditioning of the problem and an adaptive parametrization should be preferred. Observe moreover that in adaptive algorithms the Gauss Newton method is applied only to those parameters belonging to $\Lambda^{(k)}$: avoiding parameters less than the threshold ϵ_2 is useful to reduce columns linear dependence.

Moreover, as analyzed in section 6.1, at the stationary regime, the problem becomes ill-conditioned: thus, considering only the transitional regime, time localization could limit the ill-conditioning of the problem. This is evident e.g. in figure 14, where the four algorithms are compared: without time localization (red dotted line) the condition number of the sensitivity matrix has a much higher upper bound.

6.6. Sensitivity of the fourth algorithm to thresholds variations

It is interesting to analyze what happens when thresholds used in the fourth algorithm are changed. ϵ_1 decides when a the segment corresponding to a parameter should be refined: it is important to keep it not too low, to avoid over-refinements. ϵ_2 is such that parameters less than it are not considered to build the sensitivity matrix: avoiding small parameters reduces computational cost and the ill-conditioning of the problem, since we expect that they are not effective in output variations.

Previous observations are summarized in table 4, where test 1 is considered to understand how convergence results varies when thresholds are slightly changed: when ϵ_1 is decreased the over-refinement increases, while when ϵ_2 is lower both the computational cost (number of iterations) and the condition number increase. When both ϵ_1 and ϵ_2 decrease both the distance from the optimal subdivision and the computational cost and the condition number increase. Thus in general to reduce the cost is it better to increase ϵ_1, while to obtain more accurate results it could be useful to adopt smaller ϵ_1 and ϵ_2.

7. Conclusions

This paper presents a mathematical algorithm to solve a class of parabolic inverse problems based upon a convection-diffusion-reaction equation, extending some ideas presented in [8] and [22]. Both liquid (e.g. water) and gas (e.g. air) pollution problems could be considered: when source location is known, we have demonstrated that the problem is well-posed and can be solved e.g. using the Projected damped Gauss Newton method. When Γ_{in} is unknown, we have proved that adaptive parametrization with time localization is an effective strategy to estimate a sparse vector of parameters.

It could be interesting to introduce also an unrefinement strategy, trying to get closer to the optimal subdivision. For example consider figure 15: the optimal strategy would estimate only one parameter in $[1,2] \times \{1\}$, and it would not bisect the segment $[1,2]$. Instead algorithm 4 bisects $[1,2]$: the problem here is that the direction of the convective field $\mathbf{u}$ produces an overestimate of the right hand side parameter of $[1,2]$ and an underestimate of the left hand side one.

Another interesting aspect could be the generalizzation of the problem to time varying boundary conditions on Γ_{in} and to analyze more deeply the problem when space-time varying velocity fields are considered.

Acknowledgments

The authors are grateful to P. Mannucci for helpful discussions and her fundamental contribution in the demonstration of Lemma 5.1.

Appendix A. The importance of stabilizing the problem

Dealing with convection dominated problems ($\|\mathbf{u}\| >> \mu$) could be problematic, due to spurious oscillations caused by the standard FE method. The simplest way to stabilize the problem is to refine the mesh, i.e. to consider a higher number of degrees of freedom; otherwise on a coarse mesh a stabilization method such as SUPG, DW or GLS, to mention only some of them, should be used. To simplify the problem in the following we apply the simplest strategy, i.e. we refine the mesh. However a stabilization method could be included in the model, modifying the weak FE formulation. Stabilization techniques are used e.g. in [4, 6].

In this section we want to point out that the problem must be stabilized to obtain a correct estimate. In fact consider $\Omega = [0,8] \times [0,1]$, $\Gamma_h = [0,8] \times \{1\} \cup [0,8] \times \{0\}$, the velocity field $\mathbf{u}$ is modelled as a Poiseuille flow i.e.

$$\mathbf{u}(x_1, x_2) = \begin{pmatrix} -4\nu x_2^2 + 4\nu x_2 \\ 0 \end{pmatrix},$$

assume moreover that $\mu = 0.1$, $\sigma = 0.1$, $C_{up} = 0.1$ and $\Gamma_{in} = [0.5, 1] \times \{1\}$, $\vartheta = 100$. Apply to it the adaptive strategy with time localization, on different meshes. Results are depicted in figure A.16. As it can be seen, when the mesh is too coarse, the presence of spurious oscillations compromise the convergence of the algorithm to the real profile, whereas adopting a fine mesh eliminates them and gives a good estimate of the boundary control.

References

[1] A.C. Antoulas, "Approximation of large-scale dynamical systems", Siam, 2005

[2] O.M. Alifanov, E.A.Artyukhin, S.V. Rumyantsev, ”Extreme Methods for Solving III-Posed Problems with Applications to Inverse Heat Conduction Problems”, Begell House, 1995

[3] C.A. Aster, B.Borchers, C.H.Thurber, ”Parameter Estimation and Inverse Problems”, Elsevier, 2005

[4] R. Becher, B.Vexler, “Optimal control of the convection-diffusion equation using stabilized finite element methods”, *Numerische Mathematik*, **106**, 2007, 349-367

[5] R. Bracconier, F. Bonthoux, “A Numerical Method of Reconstructing the Pollutant Concentration Field in a Ventilated Room”, *Ann.Occup.Hyg.*, **51**, 2007, 311-325

[6] S. S. Collis, M. Heinkenschloss “Analysis of the streamline upwind/petrov galerkin method applied to the solution of optimal control problems”, CAAM TR02-01, 2002

[7] X. Davoine, M. Bocquet, “Inverse modelling-based reconstruction of the Chernobyl source term available for long-range transport”, *Atmos.Chem.Phys.*, **7**, 2007, 1549-1564

[8] G.Deolmi, F.Marcuzzi, S.Marinetti, S.Poles “Numerical algorithms for an inverse problem of corrosion detection”, *Communications in Alliped and Industrial Mathematics*, **1**, 2010, 78-98

[9] A. Hamdi, “Identification of Point Sources in Two Dimensional Advection-Diffusion-Reaction Equation: Application to Pollution Sources in a River. Stationary Case”, *Inverse Problems in Science and Engineering*, **00**, 2006, 1-20

[10] A. Friedman, ”Partial Differential Equations of Parabolic Type”, Dover Publications, 2008

[11] M.Girault, D.Maillet, F.Bonthoux, B. Galland, P. Martin, R. Braconnier, J. R. Fontaine, “Estimation of time-varying pollutant emission rates in a ventilated enclosure: inversion of a reduced model obtained by experimental application of the modal identification method”, *Inverse Problems*, **24**, 2008, 1-22

[12] M. Girault, D.Petit, ”Resolution of linear inverse forced convection problems using model reduction by the modal identification method: application to turbulent flow in parallel-plate duct”, *Int. J. Heat Mass Transfer*, **47**, 2004, 3909-3925

[13] M.D. Gunzburger, J.S. Peterson, J.N.Shadid ”Reduced-order modeling of time-dependent PDEs with multiple parameters in the boundary data”’, *Comput. Methods Appl. Mech. Engrg.* **196** (2007) 1030-1047

[14] M.Hinze, S.Volkwein, "Proper Orthogonal Decomposition Surrogate Models for Nonlinear Dynamical Systems : Error Estimates and Suboptimal Control", *Lecture Notes in Computational Science and Engineering*, **45**, 2005

[15] V.Isakov, "Inverse Problems for Partial Differential Equations", Springer, 2006

[16] B. Kaltenbacher, A. Neubauer, O. Scherzer "Iterative Regularization Methods for Nonlinear Ill-Posed Problems", Walter de Gruyter, 2008

[17] A.Kirsch, "An introduction to the mathematical theory of Inverse Problems", Springer, 1996

[18] K. Kunisch, S. Volkwein "Galerkin proper orthogonal decomposition methods for parabolic problems"', *Numer.Math.* **90** (2001) 117-148

[19] O.A. Ladyzenskaja, V.A. Solonnikov, N.N. Uralceva "Linear and Quasilinear Equations of Parabolic type", American Mathematical Society, 1968

[20] G. Lube, B. Tews,"Distributed and boundary control of singularly perturbed advection-diffusion-reaction problems", *Lecture Notes in Computational Science and Engineering*, **69**, 2009, 205-215

[21] F. Marcuzzi,S. Marinetti, "Efficient reconstruction of corrosion profiles by infrared thermography", *Journal of Physics: Conference Series*, **124**, no. 012033, 2008

[22] F. Marcuzzi, "Space and time localization for the estimation of distributed parameters in a finite element model", *Computer Methods in Applied Mechanics and Engeneering*, **198**, 2009, 3020-3025

[23] J.R.R.A. Martins, P.Sturdza, J.J.Alonso "The Complex-Step Derivative Approximation", *ACM Transactions on Mathematical Software*, **29**, 2003, 245-262

[24] B.P. McGrail, "Inverse reactive transport simulator (INVERTS): an inverse model for contaminant transport with nonlinear adsorption and source terms", *Environ. Model. Softw.*, **16**, 2001, 711-723

[25] A.Moutsoglou, "An inverse convection problem", *J. Heat Transfer*, **111**, 1989, 37-43

[26] J.Nocedal, S.J. Wright, "Numerical optimization", Springer, 1999

[27] H.M.Park, J.Chung, "A sequential method of solving inverse natural convection problems", *Inverse Problems*, **18**, 2002, 529-546

[28] A.Quarteroni, A.Valli, "Numerical approximation of Partial Differential Equations", Springer, 1994

[29] A. Rap, L. Elliott, D.B.Ingham, D. Lesnic, X.Wen,"An inverse source problem for the convection-diffusion equation", *International Journal of Numerical Methods for Heat & Fluid Flow*, **16**,2006, 125-150

[30] J.R. Shenefelt, R. Luck, R.P. Taylor, J.T. Berry, "Solution to inverse heat conduction problems employing singular value decomposition and model-reduction", *Int. J. Heat Mass Transfer*, **45**,2002, 67-74

[31] W.H.A. Schilders, H.A. van der Vorst, J. Rommes, "Model Order Reduction: Theory, Research Aspects and Applications", Springer, 2008

[32] A.N. Tikhonov, V.Y. Arsenin, "Solutions of Ill-Posed Problems", V HWinston, 1977

[33] K.A. Woodbury, "Inverse Engineering Handbook", CRC Press, 2002

[34] N. Zabaras, G. Z. Yang, "A functional optimization formulation and implementation of an inverse natural convection problem", *Comput. Methods Appl. Mech. Engrg.*, **144**, 1997, 245-274

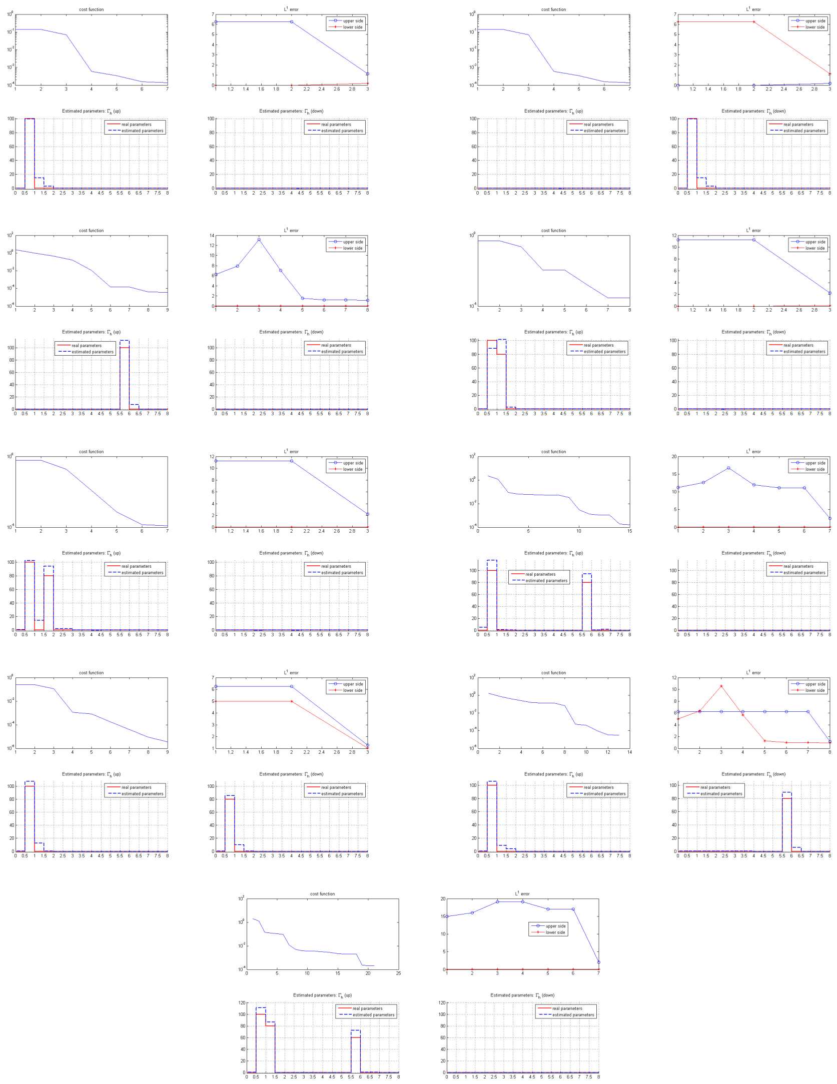
Figure 9: *Nine test cases: results of the adaptive strategy with time localization: computed estimate (blu dotted line), real control (red line). For each figure: cost function (first row, left), L^1 error (first row, right), approximation of the upper horizontal segment (second row, left), approximation of the bottom horizontal segment (second row, right).*

Test		Finest subdivision			Finest subdivision + time localization			Adaptive subdivision			Adaptive subdivision + time localization		
		up	*down*		*up*	*down*		*up*	*down*		*up*	*down*	
1	L^1-*err*	10^{-12}	10^{-12}		0.442	0.442		7.69	0.12		1.15	0.168	
	opt. sub.	+11	+14		+11	+14		+1	0		+1	0	
	num. it.			2			18			4			7
	$\mathcal{F}_d(\boldsymbol{\vartheta})$			10^{-20}			10^{-6}			10^{-5}			10^{-4}
2	L^1-*err*	10^{-12}	10^{-12}		0.02	0.02		0.12	7.69		0.168	1.15	
	opt. sub.	+14	+11		+14	+11		0	+1		0	+1	
	num. it.			2			20			4			7
	$\mathcal{F}_d(\boldsymbol{\vartheta})$			10^{-20}			10^{-6}			10^{-5}			10^{-4}
3	L^1-*err*	2.72	0.3974		0	0		1.33	0.02		0.18	10^{-3}	
	opt. sub.	+11	+14		+11	+14		0	+1		+2	0	
	num. it.			6			17			9			9
	$\mathcal{F}_d(\boldsymbol{\vartheta})$			0.028			10^{-20}			0.0012			10^{-5}
4	L^1-*err*	8.911	0.1102		1.021	10^{-3}		11.25	0.16		2.247	0.07	
	opt. sub.	+10	+14		+10	+14		0	0		0	+1	
	num. it.			5			13			4			8
	$\mathcal{F}_d(\boldsymbol{\vartheta})$			10^{-3}			10^{-3}			10^{-3}			10^{-5}
5	L^1-*err*	5.576	0.047		1.611	10^{-4}		12	0.14		2.224	0.03	
	opt. sub.	+10	+14		+10	+14		-1	0		+1	+1	
	num. it.			5			11			3			7
	$\mathcal{F}_d(\boldsymbol{\vartheta})$			10^{-3}			10^{-4}			10^{-4}			10^{-5}
6	L^1-*err*	2.653	0.2871		8.591	10^{-13}		2.33	0.01		2.48	0.01	
	opt. sub.	+8	+14		+8	+14		+1	0		+3	0	
	num. it.			5			17			10			15
	$\mathcal{F}_d(\boldsymbol{\vartheta})$			10^{-2}			0.068			10^{-4}			10^{-4}
7	L^1-*err*	10^{-13}	10^{-13}		0.36	0.36		7.63	6.12		1.267	1.019	
	opt. sub.	+9	+9		+9	+9		0	0		+1	+1	
	num. it.			2			17			4			9
	$\mathcal{F}_d(\boldsymbol{\vartheta})$			10^{-20}			10^{-6}			10^{-5}			10^{-6}
8	L^1-*err*	1.969	1.002		6.25	9.17		14.9	8.9		0.95	0.95	
	opt. sub.	+9	+9		+9	+9		+2	0		+1	+2	
	num. it.			5			17			5			13
	$\mathcal{F}_d(\boldsymbol{\vartheta})$			10^{-2}			0.13			0.19			10^{-5}
9	L^1-*err*	14.22	0.1818		14.34	10^{-12}		2.65	0.9		2.01	0.004	
	opt. sub.	+7	+14		+7	+14		+3	0		+2	0	
	num. it.			11			19			16			21
	$\mathcal{F}_d(\boldsymbol{\vartheta})$			10^{-2}			0.2			0.001			10^{-4}

Table 3: *Comparison between four algorithms: using the finest subdivision and the projected damped Gauss Newton method, using the finest subdivision and the localization in time, using the adaptive parametrization and using the adaptive parametrization and time localization. L^1-error in the upper and lower horizontal segments, number of points added to the optimal subdivision in the upper and lower horizontal segments, number of iterations and final cost function.*

ϵ_1	ϵ_2	L^1 error:		opt. sub.:		$\mathcal{F}_d(\boldsymbol{\vartheta})$	num. it.	mean condition number of Ψ
		up	down	up	down			
0.4	0.4	1.15	0.168	+1	0	10^{-4}	7	79.9513
0.3	0.4	1.15	0.168	+1	+1	10^{-4}	7	79.9513
0.01	0.4	1.15	0.168	+1	+7	10^{-5}	7	79.9513
0.4	0.3	1.192	0.02	+1	0	10^{-5}	8	173.2498
0.4	0.01	1.207	0.05	+1	0	10^{-6}	9	252.7891
0.01	0.01	1.259	0.01	+1	+3	10^{-6}	9	210.4405

Table 4: *Test 1: results for different values of ϵ_1 and ϵ_2.*

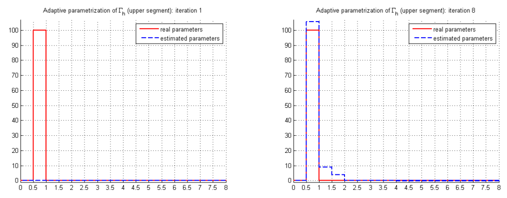

Figure 10: *Test 8. Adaptive parametrization and time localization. Evolution of the approximation (blu dotted line), real control (red line). Upper horizontal segment.*

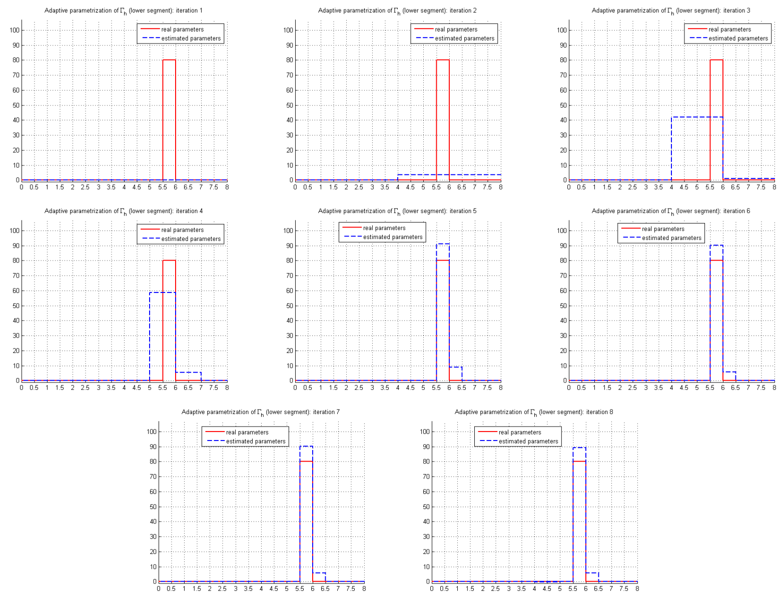

Figure 11: *Test 8. Adaptive parametrization and time localization. Evolution of the approximation (blu dotted line), real control (red line). Bottom horizontal segment.*

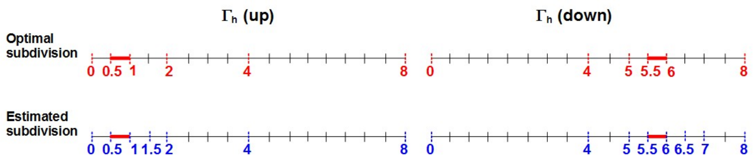

Figure 12: *Test 8. First row: optimal subdivision that could be obtained using a bisection strategy. Second row: estimated subdivision.*

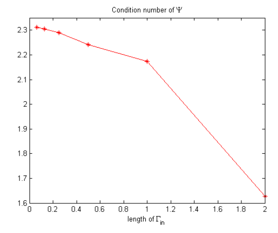

Figure 13: *Example 2, with* $\Gamma_{in} = [5 - h, 5] \times \{1\} \cup [2 - h, 2] \times \{0\}$. *Condition number of* $\Psi_{\hat{\boldsymbol{\vartheta}}}$ *for different values of* $h = \{0.0625, 0.125, 0.25, 0.5, 1, 2\}$.

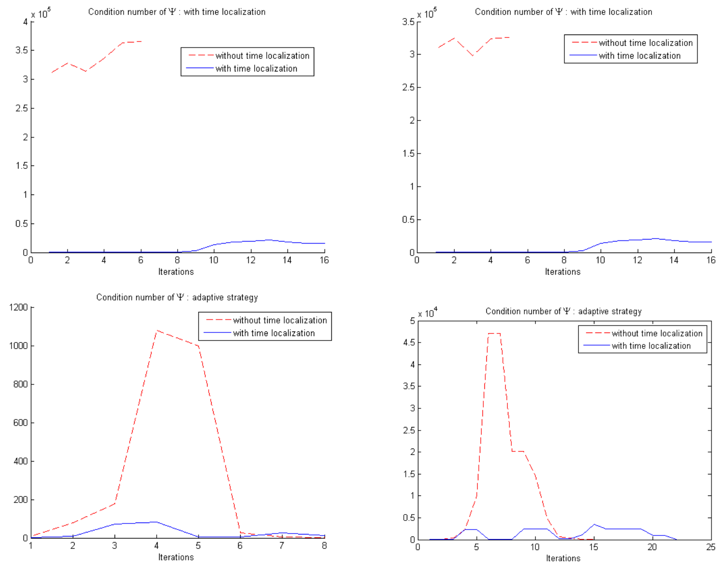

Figure 14: *Condition number of the sensitivity matrix with (blue line) and without (red dotted line) time localization. First row: finest subdivision. Second row: adaptive parametrization. Left: test 3. Right: test 9.*

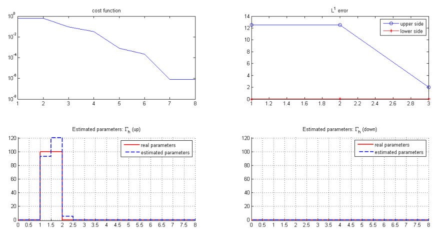

Figure 15: *Need of an under-refinement strategy.*

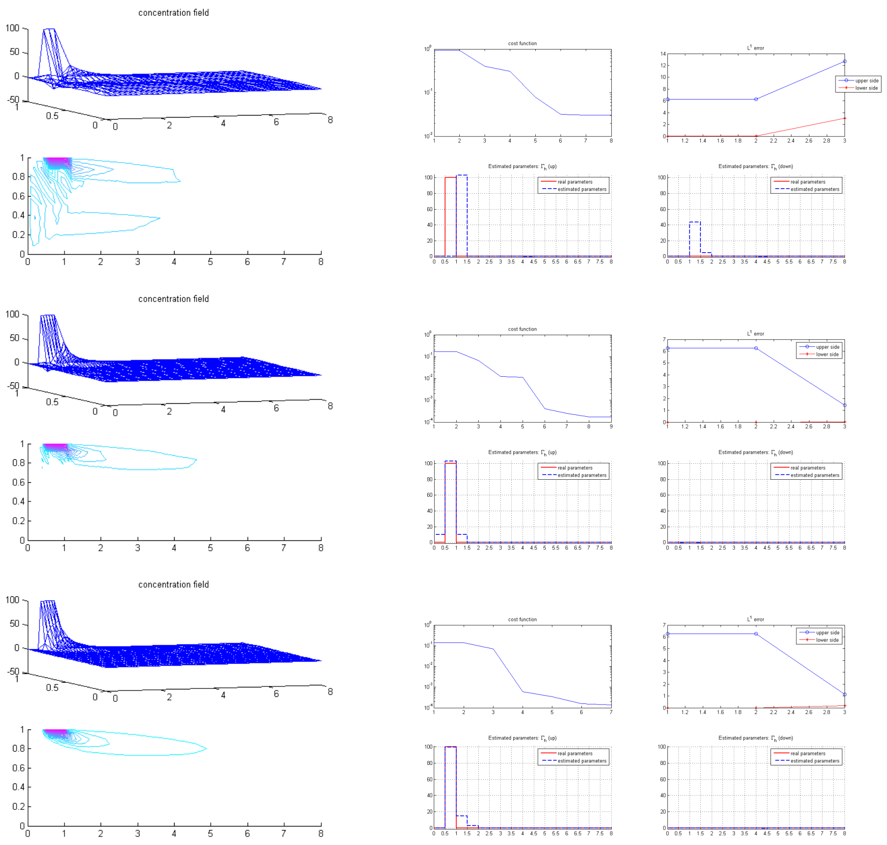

Figure A.16: *Importance of using stabilization: concentration field (left), estimated profile (right). First row: using* 41 *nodes along x-axis and* 9 *along y-axis. Second row: using* 81 *nodes along x-axis and* 13 *along y-axis. Third row: using* 81 *nodes along x-axis and* 21 *along y-axis.*